



گزارش فاز دوم پروژه

درس معماری کامپیوتر

استاد: دکتر حمید سربازی آزاد

اعضای گروه:

پویا یوسفی (۹۸۱۷۱۲۲۳)

سجاد پاکسیما (۹۸۱۰۶۲۸۶)

محمدرضا احمدی تشنیزی (۹۸۱۷۰۶۴۶)

مهدیه ابراهیم‌پور (۹۸۱۷۰۶۲۴)

نیمسال بهار ۱۴۰۱

مقدمه

در این فاز، برخلاف فاز اول حافظه اصلی در یک کلاک پاسخ را آماده نمی‌کند و به ۴ کلاک زمان نیاز دارد. از این جهت، ابتدا در پردازنده تغییراتی اعمال کردیم که در زمان اجرای دستوره‌های حافظه، به مقدار کافی تاخیر ایجاد شود؛ همچنین در این فاز یک حافظه پنهان (cache) پیاده‌سازی شده تا تاخیرات به حداقل برسند.

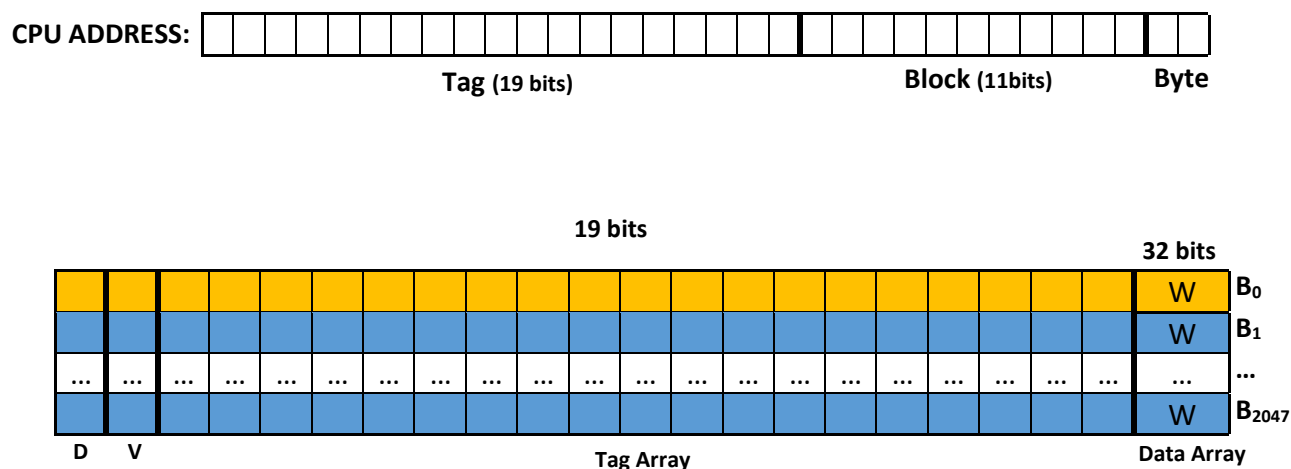
شرح کار

توقف به هنگام اجرای دستورات حافظه

برای توقف کردن، کافی است در هنگامی که تغییرات بر روی PC اعمال می‌شود یک شرط جدید تعریف کنیم. در این شرط، اگر دستور از نوع لود یا استور (lw,lb,sw,sb) بود و $hit=0$ ، آنگاه آدرس PC تغییر نمی‌کند تا عملیات لود یا استور به طور کامل انجام گیرد.

حافظہ پنہان

حافظه پنهان استفاده شده از سیاست نگاشت Direct mapped استفاده می‌کند. اندازه حافظه ۸ کیلوبایت است و هر بلوک شامل یک word می‌باشد. در نتیجه ۲۰۴۸ بلوک داریم. آدرس ورودی حافظه (CPU Address) که شامل ۳۲ بیت است، از ۳ قسمت تشکیل شده است: ۲ بیت اول برای مشخص کردن بایت مورد نظر، ۱۱ بیت بعدی برای مشخص کردن بلوک و ۱۹ بیت باقی‌مانده به عنوان آیدی بلوک یا tag استفاده می‌شود. همچنین به یک بیت validation، (V)، و یک بیت dirty، (D)، هم نیاز داریم.



ابتدا یک سیگنال کنترلی به نام cache_en به Control Unit اضافه می‌شود. این سیگنال وظیفه روشن کردن حافظه پنهان را دارد. اگر این سیگنال وجود نداشته باشد، آنگاه تمام حافظه پنهان شروع به کار می‌کند و کارکرد پردازنده را مختل می‌کند. هنگام استفاده از cache، این سیگنال روشن می‌شود و پس از آنکه کار cache تمام شود ($hit=1$) این سیگنال خاموش می‌شود. در خود ماژول Cache، ابتدا برای حالت اولیه تمام مقادیر را صفر می‌کنیم (validation, cache_memory و...). با هر کلاک، اگر سیگنال کنترلی cache روشن بود، آنگاه کار حافظه پنهان مرحله به مرحله انجام می‌شود و به کمک case، به اندازه کافی تاخیر ایجاد می‌کنیم.

ابتدا با استفاده از ۱۱ بیت بلوک، بلوک مورد نظر را پیدا می‌کنیم. سپس در صورتیکه تگ آدرس با تگ بلوک برابر بود و $valid_data = 1$ شد، اطلاعات مدنظر مطابقت دارند و بدون توقف، اطلاعات به CPU یا mips_core داده می‌شوند. اگر مقدار تگ متفاوت بود، یعنی داده مدنظر ما وجود ندارد و باید در کش ذخیره شود. اگر $dirty=0$ بود، یعنی این اطلاعات در حافظه اصلی وجود دارد. پس می‌توانیم به راحتی آن را حذف کنیم و ۴ کلاک طول می‌کشد که آدرس در حافظه پنهان ذخیره شود. اما اگر $dirty=1$ باشد، یعنی این داده در حافظه اصلی ذخیره نمی‌شود. ابتدا باید این اطلاعات به حافظه اصلی وارد شود و در آن ذخیره شود، سپس داده‌ای که CPU به cache می‌دهد در cache ذخیره شود. کل این عملیات مجموعاً ۱۱ کلاک طول می‌کشد.